Online Appendix to Background Music Recommendation on Short Video Sharing Platforms

Jiawei Chen
School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai, China, 200433, chenjiawei@mail.shufe.edu.cn

Luo He
School of Economics and Management, Tsinghua University, Beijing, China, 100084, heluo240808@163.com

Hongyan Liu
School of Economics and Management, Tsinghua University, Beijing, China, 100084, liuhy@sem.tsinghua.edu.cn

Yinghui (Catherine) Yang
Graduate School of Management, University of California, Davis, California, 95616, yiyang@ucdavis.edu

Xuan Bi
Carlson School of Management, University of Minnesota, Minnesota, 55455, xbi@umn.edu

Appendix A. Proof of TF's Ineffectiveness in Our Setting

The background music recommendation problem we study differs from traditional recommendation problems because it considers not only users and music clips (items) but also videos. A related problem is context-aware recommendation, where a context corresponds to many users and many items. In contrast, a video in our setting only corresponds to a unique user and a unique music clip. Due to this difference, classical tensor factorization models (Tucker 1966, Carroll and Chang 1970, Harshman et al. 1970), often used in recommendation systems (Frolov and Oseledets 2017) to capture relationships among users, items, and contexts (Nanopoulos et al. 2009, Bi et al. 2018), become ineffective in representing users, videos, and music clips.

There are two main families of classical tensor factorization models, namely, the family of Tucker decomposition (Tucker 1966) and the family of CP (Canonical Polyadic) decomposition (Carroll and Chang 1970, Harshman et al. 1970). We take Tucker decomposition as an example for further demonstration, and similar logic can be applied to the CP family.

For the decision of a user selecting a music clip for a video $\mathcal{C} = \{c_{i,j,k}\}$, a third-order Tucker decomposition can be generally defined as $\mathcal{C} \approx \mathcal{W} \times \mathbf{f}^u \times \mathbf{f}^v \times \mathbf{f}^m$, where $\mathcal{W} \in \mathbb{R}^{d_u \times d_v \times d_m}$ is the core tensor, and $\mathbf{f}^u \in \mathbb{R}^{I \times d_u}$, $\mathbf{f}^v \in \mathbb{R}^{J \times d_v}$, and $\mathbf{f}^m \in \mathbb{R}^{K \times d_m}$ are latent factor matrices for users, videos,

and music clips, respectively. Thus, $c_{i,j,k}$ is factorized as $\sum_{m=1}^{d_u}\sum_{n=1}^{d_v}\sum_{l=1}^{d_m}(w_{i,j,k}\cdot f^u_{i,m}\cdot f^v_{j,n}\cdot f^m_{k,l})$ and the optimization problem is written as:

$$\min_{\mathcal{W},\mathbf{f}^u,\mathbf{f}^v,\mathbf{f}^m} J=\frac{1}{2}\sum_{(i,j,k)\in S}(c_{i,j,k}-\sum_{m=1}^{d_u}\sum_{n=1}^{d_v}\sum_{l=1}^{d_m}(w_{i,j,k}\cdot f^u_{i,m}\cdot f^v_{j,n}\cdot f^m_{k,l}))^2 \tag{A.1}$$

By taking the derivative of J with respect to $f^v_{j,n}$ we can obtain:

$$\frac{\partial J}{\partial f^v_{j,n}}=-\sum_{i,k:(i,j,k)\in S}(c_{i,j,k}-\sum_{m=1}^{d_u}\sum_{n=1}^{d_v}\sum_{l=1}^{d_m}(w_{i,j,k}\cdot f^u_{i,m}\cdot f^v_{j,n}\cdot f^m_{k,l}))\sum_{m=1}^{d_u}\sum_{l=1}^{d_m}(w_{i,j,k}\cdot f^u_{i,m}\cdot f^m_{k,l}). \tag{A.2}$$

Since a video v_j only corresponds to a user u_{i_j} and a music clip m_{k_j}, Equation (A.2) can be reduced to:

$$\frac{\partial J}{\partial f^v_{j,n}}=-(c_{i_j,j,k_j}-\sum_{m=1}^{d_u}\sum_{n=1}^{d_v}\sum_{l=1}^{d_m}(w_{i_j,j,k_j}\cdot f^u_{i_j,m}\cdot f^v_{j,n}\cdot f^m_{k_j,l}))\sum_{m=1}^{d_u}\sum_{l=1}^{d_m}(w_{i_j,j,k_j}\cdot f^u_{i_j,m}\cdot f^m_{k_j,l}). \tag{A.3}$$

We take this derivative as zero and obtain:

$$\sum_{n=1}^{d_v} f^v_{j,n}=\frac{c_{i_j,j,k_j}}{\sum_{m=1}^{d_u}\sum_{l=1}^{d_m}(w_{i_j,j,k_j}\cdot f^u_{i_j,m}\cdot f^m_{k_j,l})} \tag{A.4}$$

Once Equation (A.4) holds, the objective in Equation (A.1) is zero, which means that Tucker decomposition with more parameters than matrix factorization is an overfitted model. Latent factors ($\mathbf{f}^u$, $\mathbf{f}^v$, and $\mathbf{f}^m$) defined in Tucker decomposition become ineffective in representing users, videos, and music clips.

Appendix B. Model Establishment of TF-BGM and MF-BGM

Appendix B.1. TF-BGM

In TF-BGM, we adopt Tucker decomposition(Tucker 1966) as the backbone model. For the probability of user u_i choosing music clip m_k for video v_j, we define a third-order Tucker decomposition as follows:

$$P(c_{i,j,k}=1 \mid u_i, v_j, m_k) = \frac{1}{1+\exp(-\hat{c}_{i,j,k})}, \text{ and } \hat{\mathcal{C}} = \mathcal{W} \times \mathbf{f}^u \times \mathbf{f}^v \times \mathbf{f}^m, \tag{B.1}$$

where $\mathcal{W} \in \mathbb{R}^{d_u \times d_v \times d_m}$ is the core tensor, and $\mathbf{f}^u \in \mathbb{R}^{I \times d_u}$, $\mathbf{f}^v \in \mathbb{R}^{J \times d_v}$, and $\mathbf{f}^m \in \mathbb{R}^{K \times d_m}$ are latent factor matrices for users, videos, and music clips, respectively.

Given that each video in our setting is uniquely associated with a specific user and a distinct music clip, the introduction of video-related lower-dimensional tensors makes classical tensor factorization models ineffective, as demonstrated in Appendix A. To address this issue, we introduce a transformation layer to transform video features into video-related lower-dimensional tensors:

$$\mathbf{f}_j^v = \tanh(\mathbf{W}_v \mathbf{x}_j^v + b_v), \tag{B.2}$$

where $\mathbf{W}_v$ is the weight matrix, b_v is the bias of the linear transformation, and $\tanh(\cdot)$ acts as the activation function. With the improved video representation learning, the probability defined in Equation (B.1) can then be well estimated and predicted.

Since users' decisions over music clips are binary (namely, choose or not choose), we consider a logistic link function. We denote $\mathcal{C}^+ = \{(i,j,k) : c_{i,j,k} \text{ is observed}\}$ as the observed interaction data. Note that since one user can only pick one music clip for each video, it is very difficult to identify whether the unused music clips are not considered by the user deliberately, or the user is unaware of those music clips. Therefore, we follow the common strategy (Koren 2008, Elkahky et al. 2015, He et al. 2017, Chen et al. 2021) to sample negative samples $\mathcal{C}^-$ from unseen entries in $\mathcal{C}$ instead of considering all negative samples, which also reduces the computational cost.

We put positive samples and negative samples together and adopt the binary cross-entropy loss function to train the model:

$$L(c_{i,j,k}, \hat{P}_{c_{i,j,k}}) = -\frac{1}{|\mathcal{C}^+|} \sum_{(i,j,k)\in\mathcal{C}^+} c_{i,j,k} \log(\hat{P}_{c_{i,j,k}}) - \frac{1}{|\mathcal{C}^-|} \sum_{(i,j,k)\in\mathcal{C}^-} (1-c_{i,j,k}) \log(1-\hat{P}_{c_{i,j,k}}), \tag{B.3}$$

where $\hat{P}_{c_{i,j,k}}$ stands for $P(c_{i,j,k}=1 \mid u_i, v_j, m_k)$ in Equation (B.1).

Appendix B.2. MF-BGM

In MF-BGM, we focus on modeling user-music and video-music matching under the matrix factorization framework.

To capture user-music matching, we utilize the existing matrix factorization method to measure users' preferences for music clips. We consider a user-music interaction matrix $\mathcal{R}_{I\times K}$ extracted from $\mathcal{C}_{I\times J\times K}$. Specifically, the (i,k)-th element of $\mathcal{R}_{I\times K}$ denotes whether the i-th user has chosen the k-th music clip before. That is, element $r_{i,k}=1$ means the i-th user u_i chooses the k-th music clip m_k, and 0 otherwise. We assign user u_i a latent factor vector $\mathbf{p}_i$ and music clip m_k a latent factor vector $\mathbf{q}_k$. The elements in $\mathbf{p}_i$ represents latent characteristics of user u_i in terms of her taste towards different latent factors of music clips. Similarly, the elements in $\mathbf{q}_k$ represents the extent to which the music clip m_k possesses those latent factors (Koren et al. 2009). We use the following formula which generates a scalar to approximate user u_i's preference for music clip m_k:

$$\text{preference}(u_i, m_k) = \mathbf{p}_i^{\mathrm{T}}\mathbf{q}_k. \tag{B.4}$$

For video's content matching with music, since video and music features come from different feature spaces and describe different attributes of videos and music clips, we use features of a video's related music clips and features of a music clip's related videos together with the original features for videos and music clips to match videos and music clips in both video feature space and music feature space.

In the video feature space, we compare the video's features with the average features of the videos that have adopted this music clip before. We use $\mathbf{x}_j^v$ to represent the features of video v_j and $\bar{\mathbf{x}}_k^v$ to represent the average features of the videos that have adopted m_k before. We then define the matching score between v_j and m_k in the video feature space as:

$$r_{j,k}^v = \mathbf{x}_j^v \bar{\mathbf{x}}_k^v. \tag{B.5}$$

Things are a little different when evaluating how well a new video v_j matches with a music clip m_k in the music feature space. This is because that this new video which is to be recommended

music clips has no past music adoption information. To get this video's related music clips, we locate the top l_v most similar videos to video v_j. The video's related music clips are the music clips used in these most similar videos. The features of the original music clip $\mathbf{x}_k^m$ are then compared with the average features of video v_j's related music clips $\bar{\mathbf{x}}_j^m$. In the music feature space, we define the matching score between v_j and m_k as:

$$r_{j,k}^m = \mathbf{x}_k^m \bar{\mathbf{x}}_j^m. \tag{B.6}$$

Note that $\bar{\mathbf{x}}_j^m$ and $\mathbf{x}_k^m$ are normalized feature vectors, which means that $r_{j,k}^m$ is equivalent to the cosine similarity between $\bar{\mathbf{x}}_j^m$ and $\mathbf{x}_k^m$. This is also the case for $r_{j,k}^v$.

Thus, we obtain the video-music matching score as:

$$\text{matching}(v_j, m_k) = \beta_v r_{j,k}^v + \beta_m r_{j,k}^m \tag{B.7}$$

Note that the corresponding weights β_m and β_v are learned from the data, and can show us the relative importance of $r_{j,k}^m$ and $r_{j,k}^v$ respectively.

Combining both user-music and video-music matching above, we obtain the probability of user u_i choosing music clip m_k for video v_j:

$$P(c_{i,j,k} = 1 \mid u_i, v_j, m_k) = \frac{1}{1+\exp(-\tilde{c}_{i,j,k})}, \text{ and } \tilde{c}_{i,j,k} = \mu + b_i + b_k + \mathbf{p}_i^T \mathbf{q}_k + \beta_v r_{j,k}^v + \beta_m r_{j,k}^m, \tag{B.8}$$

where μ, b_i, b_k stand for the overall main effect, user-specific, and music-clip-specific biases, which account for effects or biases from the platform level, the user level, and the music clip level, respectively.

We also adopt the binary cross-entropy loss function to train the model:

$$L(c_{i,j,k}, \tilde{P}_{c_{i,j,k}}) = -\frac{1}{|\mathcal{C}^+|} \sum_{(i,j,k)\in\mathcal{C}^+} c_{i,j,k} \log(\tilde{P}_{c_{i,j,k}}) - \frac{1}{|\mathcal{C}^-|} \sum_{(i,j,k)\in\mathcal{C}^-} (1 - c_{i,j,k}) \log(1 - \tilde{P}_{c_{i,j,k}}), \tag{B.9}$$

where $\tilde{P}_{c_{i,j,k}}$ stands for $P(c_{i,j,k} = 1 \mid u_i, v_j, m_k)$ in Equation (B.8).

Appendix C. Full Results of Ablation Studies

In addition to HR@5 reported in Table 6 in the main text, Table C.1 also include results based on HR@10, NDCG@5, and NDCG@10. We can see that our full model DL-BGM performs better than all other models. From V2M and U2M, we can infer that adding the user-music matching module leads to a higher performance improvement which is about 70% (from 65.1% to 72.3%), while the contribution of the video-music matching module is about 20% (from 17.2% to 24.3%). User adopted music clips contribute as much as video's related music clips, as we can see the results from U2M/R+V2M and U2M+V2M/R. The attention layer for the user-music matching module has a contribution of 4.82% in average, while the attention layer for the video-music matching module has a slightly less contribution of 4.71% in average.

Table C.1 Results of ablation study for component contribution of DL-BGM.

Model	HR@5	HR@10	NDCG@5	NDCG@10
V2M	0.1136 (+72.3%)	0.1565 (+76.6%)	0.0823 (+65.1%)	0.0961 (+68.5%)
U2M/R+V2M	0.1859 (+5.22%)	0.2656 (+4.05%)	0.1295 (+4.99%)	0.1551 (+4.45%)
U2M/A+V2M	0.1870 (+4.64%)	0.2630 (+5.08%)	0.1299 (+4.67%)	0.1544 (+4.89%)
U2M	0.1594 (+22.7%)	0.2359 (+17.2%)	0.1094 (+24.3%)	0.1339 (+21.0%)
U2M+V2M/R	0.1861 (+5.11%)	0.2651 (+4.25%)	0.1295 (+5.00%)	0.1549 (+4.54%)
U2M+V2M/A	0.1865 (+4.87%)	0.2643 (+4.57%)	0.1298 (+4.75%)	0.1548 (+4.64%)
DL-BGM	0.1956	0.2764	0.1359	0.1620

In addition to HR@5 reported in Table 7 in the main text, Table C.2 also include results based on HR@10, NDCG@5, and NDCG@10. Comparing GLMP+CT, CLMT+CT, GLPT+CT, GMPT+CT, and LMPT+CT, we see that the tempo features improves HR and NDCG the most by around 6% (from 5.25% to 6.21%). Comparing MF-BGM-C and MF-BGM-T, we find that the text embeddings contribute more than the CNN features.

Table C.2 Results of ablation study for feature contribution of DL-BGM.

Model	HR@5	HR@10	NDCG@5	NDCG@10
GLMP+CT	0.1859 (+5.25%)	0.2602 (+6.21%)	0.1291 (+5.30%)	0.1531 (+5.81%)
GLMT+CT	0.1867 (+4.78%)	0.2655 (+4.09%)	0.1300 (+4.56%)	0.1554 (+4.24%)
GLPT+CT	0.1863 (+4.99%)	0.2626 (+5.22%)	0.1297 (+4.80%)	0.1543 (+4.96%)
GMPT+CT	0.1863 (+5.02%)	0.2640 (+4.68%)	0.1300 (+4.54%)	0.1550 (+4.46%)
LMPT+CT	0.1869 (+4.66%)	0.2655 (+4.10%)	0.1301 (+4.48%)	0.1554 (+4.21%)
GLMPT+C	0.1741 (+12.4%)	0.2527 (+9.34%)	0.1192 (+14.1%)	0.1445 (+12.1%)
GLMPT+T	0.1858 (+5.29%)	0.2657 (+4.00%)	0.1290 (+5.37%)	0.1548 (+4.65%)
DL-BGM	0.1956	0.2764	0.1359	0.1620

Appendix D. Cold-Start Recommendations

We conduct further evaluations on cold-start recommendations for new users with no previous history of video creation and new music clips with no history of being adopted.

In our experiments, we treat the 6,422,443 users who are in the raw dataset but not selected into the final dataset as new users, and build a new test set with their first videos. Due to computation constraints, we filter out the new users adopting at least one existing music clip in the final dataset and then select 10% of them as the test set for performance evaluation. Besides, we treat the 2,332 music clips which are in the raw dataset but filtered out of the final dataset as new music clips. We then put the videos using these music clips into the original test set. When making recommendations, we take both existing music clips and new music clips as candidate items. The statistics of the new test sets for cold-start recommendations are listed in Table D.1.

Table D.1 The statistics of the new test sets for cold-start recommendations.

	#videos	#music clips	#users	$\overline{\#\text{videos}}$ for each music clip	$\overline{\#\text{videos}}$ for each user
New users	436,084	1,717	436,084	253.98	1.0
New music clips	66,707	4,049	16,559	16.47	4.03

Table D.2 reports the results of making recommendations to new video creators based on HR@N and NDCG@N. We can see that our proposed model still outperforms all the baselines. Note that our proposed DL-BGM performs about 160% (from 147.9% to 175.9%) better than MF-BGM in making recommendations for new video creators.

Table D.2 Results of making recommendations to new video creators.

Model	HR@5	HR@10	NDCG@5	NDCG@10
Top Popular	0.00172 (+3104.0%)	0.00340 (+2008.1%)	0.00102 (+3894.5%)	0.00156 (+2857.1%)
KNN	0.00944 (+482.3%)	0.0162 (+342.0%)	0.00590 (+589.0%)	0.00807 (+470.8%)
MF	0.00546 (+906.1%)	0.0104 (+588.8%)	0.00326 (+1149.1%)	0.00484 (+852.4%)

Table D.2 Results of making recommendations to new video creators, continued.

Model	HR@5	HR@10	NDCG@5	NDCG@10
NeuMF (Logit)	0.00603 (+811.6%)	0.0121 (+492.3%)	0.00367 (+1006.7%)	0.00561 (+720.8%)
NeuMF (BPR)	0.00657 (+736.2%)	0.0123 (+484.8%)	0.00385 (+956.1%)	0.00568 (+711.3%)
LFM	0.0178 (+208.4%)	0.0273 (+162.8%)	0.0116 (+250.8%)	0.0146 (+214.8%)
PDSM	0.0197 (+178.5%)	0.0310 (+130.9%)	0.0127 (+220.3%)	0.0163 (+182.1%)
MF+LFM	0.00793 (+593.5%)	0.0133 (+439.4%)	0.00506 (+703.4%)	0.00678 (+579.6%)
NeuMF (Logit)+LFM	0.00948 (+480.0%)	0.0157 (+357.8%)	0.00620 (+555.6%)	0.00818 (+463.0%)
NeuMF (BPR)+LFM	0.00690 (+696.7%)	0.0116 (+519.7%)	0.00445 (+814.4%)	0.00594 (+675.6%)
MF+PDSM	0.00859 (+539.8%)	0.0145 (+394.7%)	0.00557 (+629.8%)	0.00746 (+517.0%)
NeuMF (Logit)+PDSM	0.0104 (+426.0%)	0.0169 (+323.5%)	0.00683 (+495.2%)	0.00891 (+417.0%)
NeuMF (BPR)+PDSM	0.00756 (+627.1%)	0.0126 (+470.3%)	0.00487 (+735.2%)	0.00647 (+611.5%)
FM-BGM	0.00641 (+757.8%)	0.0114 (+529.3%)	0.00403 (+908.8%)	0.00563 (+718.6%)
TF-BGM	0.00604 (+809.2%)	0.0119 (+501.6%)	0.00357 (+1038.8%)	0.00544 (+746.7%)
MF-BGM	0.0202 (+172.7%)	0.0289 (+147.9%)	0.0147 (+175.9%)	0.0175 (+162.5%)
DL-BGM	0.0550	0.0717	0.0407	0.0461

As displayed in Table D.3, the significant superiority of our proposed model over all the baselines still holds. Note that DL-BGM performs about 25% (from 18.4% to 30.6%) better than MF-BGM in making recommendations containing new music clips.

Table D.3 Results of making recommendations for new music clips.

Model	HR@5	HR@10	NDCG@5	NDCG@10
Top Popular	0.00492 (+3694.1%)	0.0156 (+1594.8%)	0.00247 (+5126.7%)	0.00583 (+2545.5%)
KNN	0.0807 (+131.2%)	0.1070 (+146.9%)	0.0611 (+111.8%)	0.0696 (+121.9%)
MF	0.0669 (+178.7%)	0.1160 (+127.7%)	0.0406 (+218.6%)	0.0563 (+173.9%)
NeuMF (Logit)	0.0579 (+222.3%)	0.1105 (+139.1%)	0.0338 (+282.7%)	0.0506 (+204.9%)
NeuMF (BPR)	0.0675 (+176.2%)	0.1217 (+117.2%)	0.0412 (+214.2%)	0.0585 (+163.9%)
LFM	0.0536 (+248.1%)	0.0887 (+197.9%)	0.0343 (+277.4%)	0.0455 (+239.2%)
PDSM	0.0528 (+253.5%)	0.0879 (+200.7%)	0.0335 (+286.3%)	0.0448 (+244.9%)
MF+LFM	0.0733 (+154.5%)	0.1249 (+111.6%)	0.0462 (+180.1%)	0.0627 (+146.3%)
NeuMF (Logit)+LFM	0.0822 (+127.0%)	0.1380 (+91.5%)	0.0518 (+149.9%)	0.0697 (+121.6%)
NeuMF (BPR)+LFM	0.0794 (+134.9%)	0.1338 (+97.5%)	0.0503 (+157.3%)	0.0677 (+128.0%)
MF+PDSM	0.0750 (+148.6%)	0.1250 (+111.4%)	0.0471 (+174.6%)	0.0631 (+144.6%)
NeuMF (Logit)+PDSM	0.0832 (+124.1%)	0.1392 (+89.8%)	0.0524 (+146.7%)	0.0704 (+119.4%)
NeuMF (BPR)+PDSM	0.0807 (+131.1%)	0.1348 (+96.0%)	0.0510 (+153.8%)	0.0683 (+126.0%)
FM-BGM	0.1054 (+77.1%)	0.1634 (+61.7%)	0.0699 (+85.1%)	0.0885 (+74.4%)
TF-BGM	0.0764 (+144.1%)	0.1427 (+85.2%)	0.0464 (+178.9%)	0.0675 (+128.5%)
MF-BGM	0.1489 (+25.3%)	0.2231 (+18.4%)	0.0990 (+30.6%)	0.1228 (+25.7%)
DL-BGM	0.1866	0.2642	0.1293	0.1543

Appendix E. Detailed Results of Generalizability Check

Appendix E.1. Generalizability across Datasets in Different Density Levels

In our main experiments, we use a dense dataset where the number of videos for each music clip and the number of videos for each user is no smaller than a threshold of ten. To examine the sensitivity of the threshold to the superiority of our proposed model, we vary the threshold among {1, 3, 5, 10, 15, 20}. Table E.1 displays the statistics of datasets with varying thresholds. Due to computing constraints, we randomly downsample users in datasets with a threshold less than ten to the size when the threshold equals ten.

Table E.1 The statistics of datasets with different thresholds.

Threshold	#videos	#music clips	#users
1	22,507	2,421	16,559
3	90,207	2,438	16,559
5	159,366	2,391	16,559
10	323,843	1,717	16,559
15	174,821	1,317	6,021
20	89,806	856	2,336

For each dataset, we conduct experiments to compare our proposed model with representative baselines, including KNN, NeuMF (Logit)+PDSM (best among all two-step models), FM-BGM (best among baselines considering user, music clip, and video factors), and MF-BGM (best among all the baselines). According to results reported in Table E.2, our DL-BGM still shows considerable relative improvements over the baselines, which demonstrates the generalizability of our proposed model across datasets in different density levels.

Table E.2 Results on datasets with different thresholds.

(a) threshold = 1

Model	HR@5		HR@10		NDCG@5		NDCG@10	
KNN	0.0521	(+255.0%)	0.0703	(+181.5%)	0.0320	(+320.1%)	0.0378	(+265.9%)
NeuMF (Logit)+PDSM	0.1328	(+39.2%)	0.1667	(+18.8%)	0.0874	(+53.7%)	0.0981	(+40.9%)
FM-BGM	0.1406	(+31.5%)	0.1589	(+24.6%)	0.1140	(+17.8%)	0.1197	(+15.5%)
MF-BGM	0.1641	(+12.7%)	0.1901	(+4.1%)	0.1261	(+6.5%)	0.1348	(+2.6%)
DL-BGM	0.1849		0.1979		0.1343		0.1383	

Table E.2 Results on datasets with different thresholds, continued.

(b) threshold = 3

Model	HR@5		HR@10		NDCG@5		NDCG@10	
KNN	0.0693	(+182.9%)	0.0896	(+187.9%)	0.0525	(+163.2%)	0.0591	(+168.1%)
NeuMF (Logit)+PDSM	0.0957	(+104.8%)	0.1523	(+69.3%)	0.0579	(+138.6%)	0.0761	(+108.0%)
FM-BGM	0.1125	(+74.2%)	0.1541	(+67.3%)	0.0822	(+68.2%)	0.0956	(+65.7%)
MF-BGM	0.1753	(+11.7%)	0.2301	(+12.0%)	0.1205	(+14.7%)	0.1381	(+14.6%)
DL-BGM	0.1959		0.2579		0.1382		0.1583	

(c) threshold = 5

Model	HR@5		HR@10		NDCG@5		NDCG@10	
KNN	0.0710	(+168.5%)	0.0912	(+184.1%)	0.0541	(+143.9%)	0.0606	(+154.2%)
NeuMF (Logit)+PDSM	0.0788	(+141.9%)	0.1347	(+92.5%)	0.0467	(+182.7%)	0.0646	(+138.3%)
FM-BGM	0.1044	(+82.5%)	0.1539	(+68.4%)	0.0710	(+85.8%)	0.0869	(+77.2%)
MF-BGM	0.1592	(+19.7%)	0.2236	(+15.9%)	0.1064	(+24.0%)	0.1271	(+21.2%)
DL-BGM	0.1906		0.2592		0.1320		0.1540	

(d) threshold = 10

Model	HR@5		HR@10		NDCG@5		NDCG@10	
KNN	0.0836	(+134.0%)	0.1109	(+149.2%)	0.0633	(+114.8%)	0.0721	(+124.7%)
NeuMF (Logit)+PDSM	0.0866	(+125.8%)	0.1440	(+91.9%)	0.0547	(+148.4%)	0.0731	(+121.5%)
FM-BGM	0.1091	(+79.2%)	0.1689	(+63.6%)	0.0724	(+87.8%)	0.0915	(+76.9%)
MF-BGM	0.1550	(+26.2%)	0.2319	(+19.2%)	0.1030	(+31.9%)	0.1277	(+26.8%)
DL-BGM	0.1956		0.2764		0.1359		0.1620	

(e) threshold = 15

Model	HR@5		HR@10		NDCG@5		NDCG@10	
KNN	0.0896	(+110.5%)	0.1192	(+125.9%)	0.0665	(+96.5%)	0.0761	(+105.9%)
NeuMF (Logit)+PDSM	0.0596	(+216.2%)	0.0972	(+177.0%)	0.0379	(+244.8%)	0.0500	(+213.5%)
FM-BGM	0.1133	(+66.5%)	0.1767	(+52.3%)	0.0734	(+78.0%)	0.0938	(+67.1%)
MF-BGM	0.1495	(+26.1%)	0.2277	(+18.2%)	0.0983	(+32.9%)	0.1235	(+26.9%)
DL-BGM	0.1885		0.2692		0.1307		0.1566	

(f) threshold = 20

Model	HR@5		HR@10		NDCG@5		NDCG@10	
KNN	0.0910	(+89.1%)	0.1270	(+97.9%)	0.0663	(+79.2%)	0.0778	(+85.3%)
NeuMF (Logit)+PDSM	0.0598	(+187.7%)	0.1011	(+148.7%)	0.0370	(+220.6%)	0.0503	(+186.9%)
FM-BGM	0.1127	(+52.7%)	0.1777	(+41.5%)	0.0724	(+64.0%)	0.0933	(+54.6%)
MF-BGM	0.1428	(+20.5%)	0.2196	(+14.5%)	0.0913	(+30.2%)	0.1159	(+24.5%)
DL-BGM	0.1721		0.2514		0.1188		0.1443	

Appendix E.2. Generalizability across Datasets in Different Categories

To further examine the cross-dataset generalizability of our proposed model, we generate new datasets by extracting videos in different categories and conduct experiments to compare with representative baselines.

On Douyin, videos cover a wide range of topics and can be grouped into many categories. Among these categories, we select seven categories that align with those present in the THU Open Chinese Lexicon (Han et al. 2016). These categories include automobiles, food, travel (location), healthcare, pets, finance and economics, and technology. For each of these selected categories, we construct a dataset comprising videos that contain text pertaining to that category. Out of the seven datasets constructed, we choose the two largest datasets: the "food-related dataset" and the "location-related dataset", in order to conduct cross-dataset generalizability assessments. The statistics of the newly organized datasets are listed in Table E.3.

Table E.3 The statistics of datasets with different video categories.

Category	#videos	#music clips	#users
Food	115,010	2,696	100,325
Location	379,770	3,222	379,770

As displayed in Table E.4, our DL-BGM still performs better than representative baselines in both the food-related dataset and the location-related dataset, which indicatess the cross-dataset generalizability of our proposed model.

Table E.4 Results on datasets with different video categories.

(a) The food-related dataset

Model	HR@5		HR@10		NDCG@5		NDCG@10	
KNN	0.0769	(+83.2%)	0.0998	(+81.2%)	0.0599	(+70.0%)	0.0672	(+70.7%)
NeuMF (Logit)+PDSM	0.0607	(+132.2%)	0.0936	(+93.3%)	0.0399	(+155.1%)	0.0504	(+127.8%)
FM-BGM	0.0923	(+52.7%)	0.1252	(+44.5%)	0.0689	(+47.7%)	0.0794	(+44.6%)
MF-BGM	0.1256	(+12.3%)	0.1563	(+15.7%)	0.0930	(+9.4%)	0.1030	(+11.5%)
DL-BGM	0.1410		0.1809		0.1018		0.1148	

Table E.4 Results on datasets with different video categories, continued.

(b) The location-related dataset

Model	HR@5		HR@10		NDCG@5		NDCG@10	
KNN	0.1194	(+75.7%)	0.1479	(+69.9%)	0.0987	(+67.9%)	0.1078	(+66.0%)
NeuMF (Logit)+PDSM	0.0908	(+131.1%)	0.1221	(+105.8%)	0.0656	(+152.8%)	0.0757	(+136.5%)
FM-BGM	0.1263	(+66.1%)	0.1450	(+73.4%)	0.1028	(+61.2%)	0.1089	(+64.4%)
MF-BGM	0.1581	(+32.7%)	0.1739	(+44.6%)	0.1291	(+28.4%)	0.1342	(+33.4%)
DL-BGM	0.2098		0.2514		0.1658		0.1790	

Appendix F. Detailed Results of Robustness Check

The HR@5 and NDCG@5 results for different numbers of similar videos are listed in Table F.1. The HR@5 results range from 0.1945 (150) to 0.1956 (100).

Table F.1 Results for different numbers of similar videos.

(a) HR@5

Model	# of similar videos				
	50	**100**	**150**	**200**	**250**
DL-BGM	0.1955	0.1956	0.1945	0.1953	0.1950

(b) NDCG@5

Model	# of similar videos				
	50	**100**	**150**	**200**	**250**
DL-BGM	0.1365	0.1359	0.1359	0.1361	0.1360

The HR@5 and NDCG@5 results for different numbers of user embedding dimensions are listed in Table F.2. The HR@5 results range from 0.1911 (100) to 0.1956 (400).

Table F.2 Results for different user embedding dimensions.

(a) HR@5

Model	User embedding dimension				
	100	**200**	**300**	**400**	**500**
DL-BGM	0.1911	0.1949	0.1952	0.1956	0.1944

(b) NDCG@5

Model	User embedding dimension				
	100	**200**	**300**	**400**	**500**
DL-BGM	0.1356	0.1367	0.1363	0.1359	0.1355

The HR@5 and NDCG@5 results for different numbers of video related dimensions are listed in Table F.3. The HR@5 results range from 0.1900 (800) to 0.1956 (2000).

Table F.3 Results for different video embedding dimensions.

(a) HR@5

Model	Video embedding dimension				
	800	**1200**	**1600**	**2000**	**2400**
DL-BGM	0.1900	0.1922	0.1953	0.1956	0.1955

(b) NDCG@5

Model	Video embedding dimension				
	800	**1200**	**1600**	**2000**	**2400**
DL-BGM	0.1308	0.1341	0.1358	0.1359	0.1358

From the above results, we can conclude that when shifting from the optimal value of parameters, the performance may have only a slight decline, which means that all the results are relatively stable and robust in all.

References

Bi X, Qu A, Shen X (2018) Multilayer tensor factorization with applications to recommender systems. *Ann. Stat.* 46(6B):3308–3333.

Carroll JD, Chang JJ (1970) Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika* 35(3):283–319.

Chen J, Yang Y, Liu H (2021) Mining bilateral reviews for online transaction prediction: A relational topic modeling approach. *Inf. Syst. Res.* 32(2):541–560.

Elkahky AM, Song Y, He X (2015) A multi-view deep learning approach for cross domain user modeling in recommendation systems. *Proc. 24th Int'l Conf. World Wide Web*, 278–288.

Frolov E, Oseledets I (2017) Tensor methods and recommender systems. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7(3):e1201.

Han S, Zhang Y, Ma Y, Tu C, Guo Z, Liu Z, Sun M (2016) THUOCL: Tsinghua Open Chinese Lexicon.

Harshman RA, et al. (1970) Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis .

He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. *Proc. 26th Int. Conf. World Wide Web*, 173–182.

Koren Y (2008) Factorization meets the neighborhood: A multifaceted collaborative filtering model. *Proc. 14th ACM SIGKDD Int'l Conf. Knowl. Discov. and Data Min.*, 426–434.

Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37.

Nanopoulos A, Rafailidis D, Symeonidis P, Manolopoulos Y (2009) Musicbox: Personalized music recommendation based on cubic analysis of social tags. *IEEE Tran. Audio, Speech, and Lang. Process.* 18(2):407–412.

Tucker LR (1966) Some mathematical notes on three-mode factor analysis. *Psychometrika* 31(3):279–311.